applied sciences

MDPI

Article

Blockchain-Based Multistage Continuous Authentication for Smart Devices

Yue Yao, Xiaomei Zhang *, Haomin Hu, Huibin Liu, Rong Huang and Zejie Wang

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; yueyao@sues.edu.cn (Y.Y.); hhm-hao@163.com (H.H.); huibinliu@sues.edu.cn (H.L.); huangrong62@163.com (R.H.); zejiewang@126.com (Z.W.)
* Correspondence: xmzhang@sues.edu.cn; Tel.: +86-21-6779-1035

Abstract: With the increasing connectivity between multiple smart devices in the Internet of Vehicles, privacy and security have become stringent threats due to unauthorized access. To overcome this issue, designing continuous authentication systems has become an important research topic because of the advantages of continuous monitoring of users after the initial access to the smart devices. Unfortunately, the existing systems are based on a third-party centralized structure, and most of them suffer storage pressure on equipment, thus resulting in significant security hazards and limited performance. In this paper, we propose a multistage continuous authentication system based on blockchain technology and the IPFS, which achieves decentralization and reduces storage pressure. In the first stage of authentication, we adopt Hyperledger Fabric to implement the underlying technical architecture of the blockchain to enhance the security and reliability of identity parameters. The preoutputs of the first-stage authentication are compared against behavioral biometric characteristics stored in the IPFS that aim to accomplish the final authentication. In particular, we use fuzzy extractors to deal with behavioral biometric feature templates, thus solving the privacy problem caused by user information leakage. To evaluate the security of our system, we prove the correctness of the communication protocol and two-way authentication of the scheme using BAN Logic. Furthermore, we use Hyperledger Caliper to analyze the impact of the sending rate of authentication requests on various performance parameters such as throughput, memory, and CPU utilization of the authentication system. Security and experimental results show that: (i) We solve the problem of centralized authentication and can resist replay attacks. (ii) Our scheme can maintain high throughput and effectively reach consensus. Compared to related works, the throughput is improved by 8.6%.

Keywords: blockchain; multistage continuous authentication; gait recognition; fuzzy extractor

check for updates

Citation: Yao, Y.; Zhang, X.; Hu, H.; Liu, H.; Huang, R.; Wang, Z. Blockchain-Based Multistage Continuous Authentication for Smart Devices. *Appl. Sci.* **2023**, *13*, 12641. https://doi.org/10.3390/app132312641

Academic Editor: Gianluca Lax

Received: 25 October 2023
Revised: 20 November 2023
Accepted: 22 November 2023
Published: 24 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid developments of communication and intelligent technologies, mobile devices have become prevalent in number in the Internet of Vehicles [1]. People prefer to store valuable personal information on smart devices for convenient storage and rapid access. Unfortunately, these devices are exposed to the leakage of private data and could be imposed by unauthorized access. Traditional access control mechanisms are based on explicit authentication methods such as passcode [2], fingerprint [3], and personal identification number (PIN) [4]. Many studies have shown the drawbacks of these one-time authentication mechanisms due to the inability to exercise adequate vigilance after initial access. To overcome the weaknesses, researchers propose continuous authentication [5] as a supplement to the existing solutions. The proposed authentication technique can continuously sense the user's interaction with the smart devices. Table 1 summarizes the distinctions between one-time authentication schemes and continuous authentication schemes. More recently, gait recognition [6] has received academic attention as an emerging biometrics technique. Moreover, each individual's gait behavior is unique and stable within

a certain time frame. As a result, gait-based authentication methods can identify users by capturing signals from how they walk.

Table 1. The distinctions between one-time authentication schemes and continuous authentication schemes.

	Antiattack Capability	Portability	Convenience	Multiple Authentication
One-time authentication	Normal	No (need to carry software, such as a smart card)	No (authentication by unlocking screen)	No (one-time)
Continuous authentication	Good	Good (embedded in smart devices)	Good (no requirement to unlock screen additionally)	Good (continuous)

Biometric information as a signing key [7] has been successfully applied in the authentication system. In fact, the gait signal can be affected by noise in the process of acquisition [8]. Moreover, there are some errors in the sampling results. It can be seen that keeping a signing key securely and accurately is a burden for the user. Therefore, how to remove noise interference is the key to realizing biometrics recognition. In 2004, Doris proposed the fuzzy extractor technique [9]. This technique can extract two biometric traits within the margin of error to generate an identical random key. This provides a security measure for the authentication process. Therefore, we propose a security model using the fuzzy extractor technique in this paper. In this security model, the fuzzy extractor extracts the gait features and generates a randomly distributed string and auxiliary public information. In the authentication phase, the same randomized string can be generated based on the auxiliary public information and biometric information, which is close to the original input. In this authentication process, the biometric real data are not stored and transmitted in any form.

Currently, common identity data storage method adopts centralized storage in a central server [10]. Nevertheless, this method has a very obvious single-point problem [11]. Moreover, authentication based on a single-server architecture is difficult to adapt to the increasingly complex network environment. To overcome these issues, extensive efforts have been devoted to decentralized technology. In 2008, Satoshi Nakamoto published a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [12]. The emergence of blockchain technology has provided new solutions for establishing distributed authentication schemes [13] since then. On the one hand, the blockchain can solve the security problems brought on by the centralized management mentioned above. On the other hand, the open and transparent characteristics of the blockchain make it possible to share information between different entities. Consequently, this paper proposes a blockchain-based multistage continuous authentication model. This model implements on-chain and off-chain authentication of user identity by combining blockchain technology and the IPFS. In addition, the IPFS can solve the limited storage problem of the blockchain. To the best of our knowledge, this is the first work to address continuous authentication with decentralization consideration via blockchain technology for smart devices.

The main contributions of this paper are summarized below:

1. We propose a multistage authentication system that combines blockchain and the IPFS. The authentication of certificate information in the blockchain is completed in the first stage of authentication. Based on the preoutput of the first stage, the user's gait feature template is checked to complete the final authentication. Compared with the traditional single-stage authentication, the proposed scheme has obvious advantages in terms of identity management, authentication, and security.
2. We present a security model using fuzzy extractor technology, which is expected to generate a randomly distributed string and auxiliary public information by extracting

gait features. This model takes full advantage of the fuzzy extractor and addresses the problems caused by biometric leakage.

3. In this paper, the blockchain relies on the Hyperledger Fabric 2.4 platform. In addition, we have built a distributed file storage system known as the IPFS to achieve the blockchain on-chain and off-chain data storage solutions. We perform security analysis and performance comparison of the scheme. The experiments show that the scheme has the advantages of high throughput and low latency. Compared to related work, the throughput of the query and update test chain reached 326 TPS and 140 TPS, respectively.

The remainder of the paper is organized as follows: we present relevant preliminaries regarding authentication and the blockchain in Section 2. Section 3 introduces the related work. In Section 4, we describe the proposed identity chain. Section 5 shows the development and experimental results of the system. Secure analysis and performance analysis are provided in Section 6. In the last section, we conclude the paper.

2. Related Work

2.1. Blockchain

Blockchain technology originated in the digital economy [14,15] and has developed in conjunction with technologies such as the Internet of Things [16] and cloud computing [17]. Initially, blockchain technology was mostly used in digital finance in order to offer more secure and cost-effective solutions for payments and crossborder transactions. Additionally, a blockchain can establish a secure data marketplace, thus making it easier for owners of IoTs devices to control the access to and sharing of their data. This not only enhances the security and credibility of data, but also brings forth increased possibilities for intelligence and automation in the Internet of Things. Integrating a blockchain with cloud computing can enhance the security of cloud computing. In particular, it can ensure the integrity of data exchanges carried out in a cloud computing environment in terms of data storage and transmission.

Blockchain serves as the underlying technology for distributed public ledger systems. It consists of blocks that hold different information. The block header mainly contains the hash of the previous block, timestamp, target hash, and Merkle root. The specific transaction information is recorded in the block body. The internal structure of the blockchain is shown in Figure 1. In particular, the operations in the blockchain are strictly timestamped [18,19] so that the transactions within a block are tightly connected.

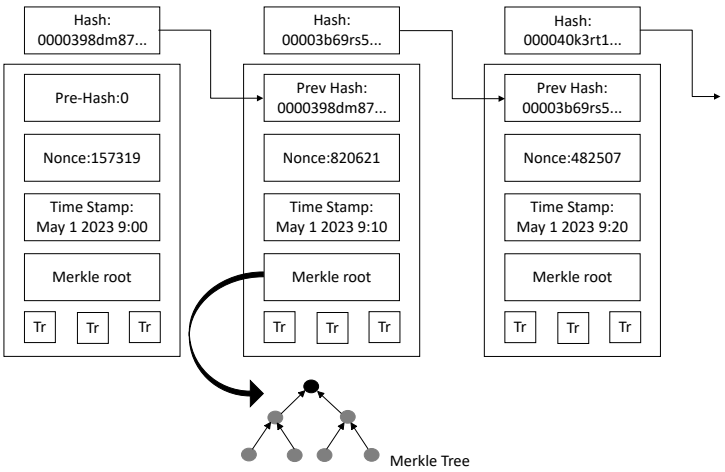


Figure 1. Blockchain architecture.

2.2. Biometric Authentication

So far, the application of authentication mechanisms in IoTs systems has been extensively explored by researchers. To improve the privacy and security of authentication

mechanisms, Tanveer et al. [20] proposed a new authentication protocol for crowdsourcing within the IoTs. This protocol enables mutual authentication between the user and the server. It also creates keys for the duration of the session. However, the authors do not mention the specific architecture used by the system. Tanveer et al. [21] proposed a resource-efficient authentication data-sharing mechanism for smart wearable systems. The security of this scheme was corroborated by operating a Scyther-based formal security analysis. However, it is restricted to the design of a lightweight structure and the secure storage of data. A secure and decentralized authentication mechanism was proposed by Petcu [22], in which software and hardware wallets can interact using a novel browser. However, the mechanism uses the Ethereum platform, and there is a risk that the account will be compromised. Ali et al. [23] designed xDBAuth, which can perform permission delegation for smart devices. However, there is no mention of how to implement two-way authentication between the server and the user. Xiong et al. [24] used smart contracts to design an identity management system that can be used to support large-scale user access. Unfortunately, the user information is transmitted in plaintext in this scheme. Table 2 summarizes the recent papers that have studied the use of biometric information for authentication techniques.

Table 2. Summary of the related works.

Research	Method	Accomplishment	Result	Drawbacks
[20]	Uses CM, AEAD algorithm, and a secure hash algorithm to design CMAP-IoTs	A new authentication protocol for crowdsourcing within IoTs	This protocol allows for mutual authentication between the user and server and establishes a session key for encrypted transmission	Cannot realize a distributed architecture and has the problem of single-point attack. Not physically deployed in IoTs devices.
[21]	Uses an authenticated encryption algorithm and hash function	An authenticated SK establishment mechanism	Provides sound protection to the patient's data and mitigates the overall cost and delay	There is a security problem with keys during long-term sessions, and it has a centralization problem.
[22]	Ethereum blockchain ecosystem	A novel browser interaction mechanism with user's software and hardware wallet	Describes a straightforward implementation of user authentication	The solution is vulnerable to phishing attacks.
[23]	Local and global smart contracts	Decentralized blockchain-based permission delegation and access control	Provides strong privacy-preserving crossdomain authentication	Does not realize formal modeling and formal verification
[24]	Ethereum blockchain ecosystem	A trust management method for reputation evaluation and a credit penalty mechanism	The scheme offers flexible and scalable trust management for large-scale IoTs	Has the problem of efficiency in large-scale IoTs devices and storage pressure
Proposed	Blockchain technology and the IPFS	A multistage continuous authentication	The first work that applies a blockchain to multistage continuous authentication.	Smart contracts and consensus algorithms for authentication can be improved

Biometric technology is a technology that uses the inherent physiological or behavioral characteristics of a person for identity authentication. In particular, gait is an emerging behavioral recognition feature that is unique between individuals and is difficult to attack through replication techniques [25–27]. Compared to biometric features such as face and iris, gait has characteristics such as infallibility and high applicability. It is gradually becoming a popular topic of interest for scholars domestically and abroad.

Identification systems based on gait generally consist of an enrollment phase and an authentication phase. The system mainly consists of the following modules: signal acquisition, data preprocessing, gait feature extraction, feature matching, and creating a gait database. These are shown in Figure 2. During the registration phase, the user's behavioral data are recorded by built-in sensors in the mobile device while the user is using the phone. The noisy behavioral data are preprocessed and fed into the feature extraction module to build a feature dataset. During the authentication phase, behavioral data are randomly collected from users. After the data preprocessing and feature extraction module, the model is used directly to authenticate the user's identity and determine whether the current user is a legitimate one.

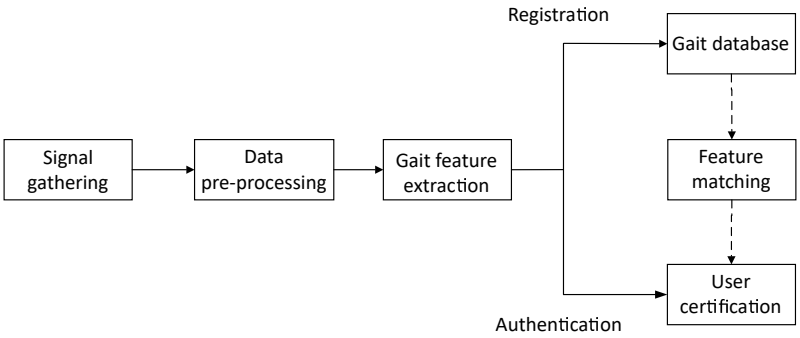


Figure 2. Authentication process.

2.3. Fuzzy Extractor

In traditional cryptography, a uniformly distributed and precisely reproducible random string is usually used as the value of the key. However, it is difficult to create and store strings in reality, and most strings cannot be reproduced randomly and reliably. For example, there is no guarantee that anyone's face or fingerprints will be consistent in each scan. As a result, Dodis [9] proposed the concept of a fuzzy extractor to convert nonuniform random source data into a reproducible uniform random string. The principle of the fuzzy extractor is shown in Figure 3. The defining formula is expressed as follows:

$$FE = (Gen, Rep) \tag{1}$$

A full fuzzy extractor consists of two stages of algorithms: The random character generation algorithm and the random character recovery algorithm. These are defined as follows:

1. Random character generation algorithm $Gen(\omega) \rightarrow ((P, R))$: The biometric information ω of the user is input, and the string R corresponding to the user's biometric is output as the user's random key and public information P;
2. Random character recovery algorithm $Rep(\omega', P) \rightarrow (R)$: Enters the user's biometric information ω' and the corresponding public information P. If the error between the two input biometrics ω and ω' is satisfied in the given permissible range ϵ, or $dis(\omega, \omega') \leq \epsilon$, then it will output the string R corresponding to the user's biometric information.

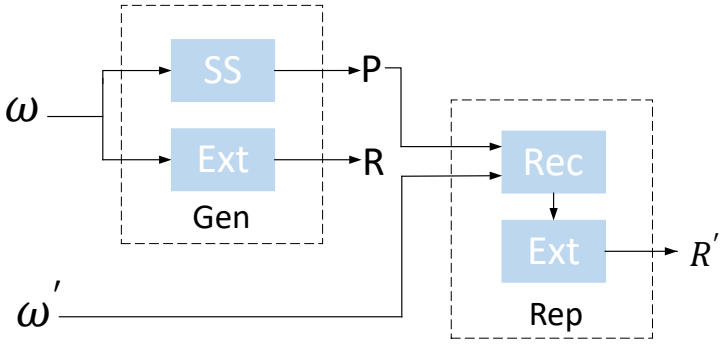


Figure 3. Fuzzy extractor.

2.4. Interplanetary File System

The IPFS is a peer-to-peer distributed file system that is implemented over a P2P network. It is designed to solve the problem of wasted storage space. When a file is added to the network, the IPFS creates a hash of the file based on its contents. The user finds the exact location of the file based on the hash code. The hash code is unique, and, therefore, the corresponding file is also unique. The hash is modified whenever the file is updated [28]. Since IPFS technology is a content-based addressing scheme, it is resistant to a certain level of security attacks. Even if there are problems with individual nodes, the system works well.

Blockchain technology and the IPFS have good performance regarding the security and scalability of data storage, respectively. However, it is expensive to use a blockchain for distributed storage. Furthermore, the storage method of the blockchain is that each node stores the full amount of data in the whole network. It is unnecessary for data storage and causes a waste of storage space. The IPFS is characterized by file deduplication and persistence. However, how to effectively manage the data stored in the IPFS is also a hard problem. Therefore, the combination of blockchain technology and the IPFS effectively reduces the storage of physical space and improves the efficiency of data processing.

3. Preliminaries

With the rapid development of computer technology, the world has entered the information age. More and more communication and interaction take place on the Internet [29–31]. Considerable amounts of personal information, as well as important information about enterprises and governments, are spread in the network. However, there would be a huge security risk once important information is intercepted or leaked. In the context of information security, it is becoming increasingly important to securely manage identities and to achieve the authentication of various identities. Before accessing the server, any user needs to be identified by the authentication system.

In order to avoid all kinds of network intrusion and malicious attacks, it is necessary to use security protection measures to ensure the security and reliability of the authentication system. In the early 1980s, Lamport et al. [32] introduced a password authentication factor in the authentication process for the first time, thus proposing an authentication scheme based on the password factor. Since then, countless experts and scholars have conducted research on identity authentication regarding the password factor and proposed many password-based authentication schemes. With the popularity of smart devices and the advancement of biometric technology, scholars have also started to try to combine biometric technology and identity authentication [33–35]. For example, Li et al. [36] proposed a new method of constructing artificial fingerprints to protect the security of fingerprint templates. It made it much more difficult for attackers to obtain real fingerprint data. Joseph et al. [37] proposed a multimodal authentication system by fusing fingerprint, iris, and palmprint feature points to provide better security for data in cloud-based environments.

In recent years, more and more scholars have conducted research on the encryption methods for biology in order to effectively balance the accuracy limitation of biometric

recognition and the accuracy requirement of keys in cryptography [38,39]. Among them, Srinivas et al. [40] proposed a new user authentication key agreement scheme using the fuzzy extractor technique, where only legitimate users can access the service using a specified IoTs device. Khan et al. [41] proposed an ECC-based mutual authentication protocol for smart grid communication. This framework uses an encrypted biometric identification approach and can satisfy a variety of security features such as replay attacks, user anonymity, and man-in-the-middle attacks. However, these common IoTs authentication schemes are mainly based on a centralized system [42–44]. That is, a trusted centralized server is used to provide authentication services for IoTs devices. However, this centralized approach to authentication is not only a potential target for malicious attacks. Moreover, centralized servers can encounter performance bottlenecks when handling large numbers of authentication requests.

In 2008, Satoshi Nakamoto [12] published a paper entitled "Bitcoin: A peer-to-peer electronic cash system". Since then, blockchain technology as the core technology of Bitcoin has received widespread attention due to its advantages of decentralization, tamper proofness, and transparency. Thus, the emergence of blockchain technology has provided a new idea for establishing a decentralized identity solution. Combining blockchain technology and identity authentication can not only solve the security problems caused by the centralized management approach mentioned above, but the open and transparent nature of the blockchain makes it possible to share information. Lee et al. [45] proposed a blockchain-based digital identity management model, BIDaaS, for identity management and authentication. In the system, users create a virtual ID to identify their digital identity, which is bound with public and private key information. Users prove their identities by certifying that they have the private key corresponding to the ID. Zhang et al. [46] also proposed a distributed identity authentication scheme using the blockchain, but the difference with [45] is that this system stores the information in a distributed storage system, while the blockchain only stores the root hash as authentication.

4. System Model

4.1. Problem Model

Authentication is the process of identifying users in order to obtain personal information and authorize their access. The traditional model of an authentication system is shown in Figure 4. The model consists of three main components: the user, the attacker, and the central server, which is maintained by the service providers. The user inputs ID and certificate information to request the service. Then, the service provider initiates the request of user identity registration or authentication to the central server. The server writes or verifies the user information and returns the result. If an attacker attempts a persistent attack to paralyze the system, the authentication system that has received a denial-of-service (DoS) attack will not be able to process the user's authentication request.

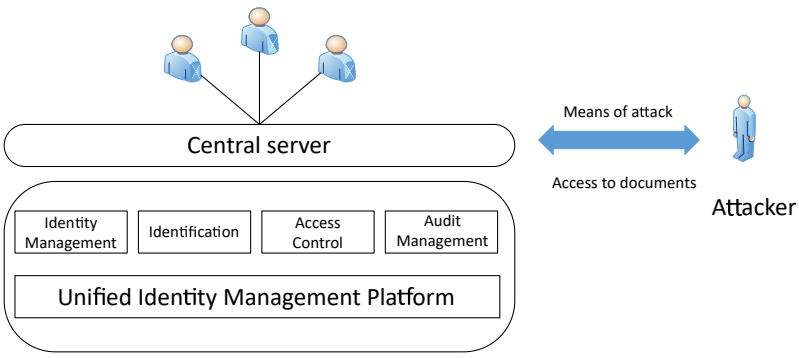


Figure 4. Problem model.

4.2. System Framework

In order to improve the problem model of the identification, this paper proposes an authentication system based on blockchain technology. The data flow of the system's storage scheme is shown in Figure 5. The modular components included are the client, service provider, IPFS, and Hyperledger Fabric:

(1) The client packages the identity data of the registered user to request the server. The server provider preprocesses the data. The identity data includes username, biometric information, and documentation information.
(2) The server provider cannot store the user data for storage; at the same time, the server itself has weak computing power. Therefore, the server provider will upload user data to the peer node in the IPFS network to realize data storage and backup. In addition, the user name and document information will also be stored in the blockchain through transactions.
(3) The IPFS creates a hash address of the uploaded biometric information and returns it to the server provider, which ensures that there are no files sharing the same hash. During the authentication process, the IPFS searches files by their content identifiers, rather than by their location.
(4) The server provider packages the username, document information, and hash address to perform the chaining operation. This scheme adopts Hyperledger Fabric to store and manage the data in the system and at the same time utilizes smart contracts to realize the interaction with the server provider.
(5) The interaction information will be fixed to the block through transactions in Fabric to realize the persistent storage of related data. The blockchain returns whether the upload has succeeded.
(6) Successful uploading means that the client has applied for the service successfully on the server provider. If the upload fails, the server returns the reason for failure to the client.

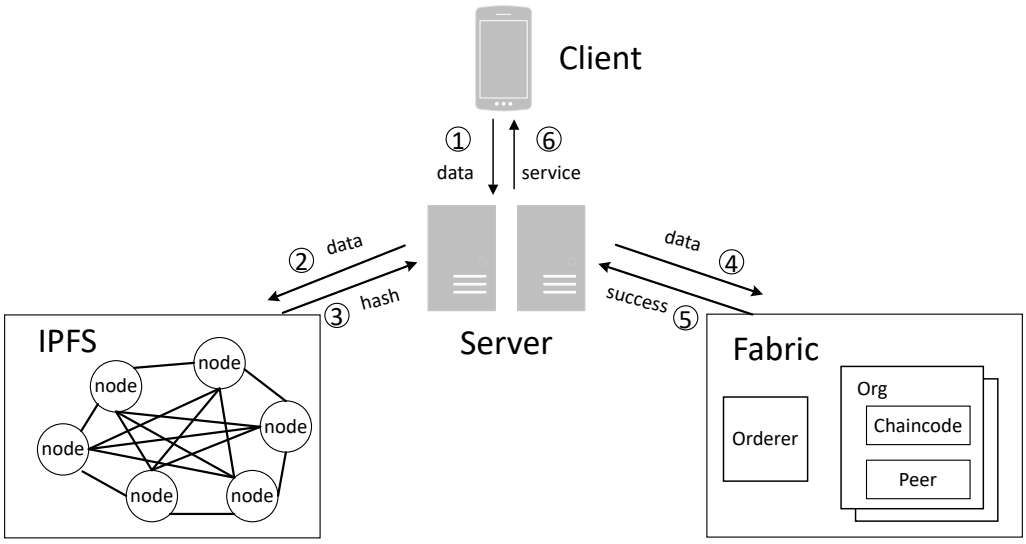


Figure 5. System model.

The specific implementation details for registration and authentication are as follows. The framework of the identity registration model based on the blockchain is shown in Figure 6. The client filters the optimal combination ω_i. The fuzzy extractor receives the biometric vector ω_i and generates the user's random key R_i and public information P_i using a random character generation algorithm. It returns the random key R_i to the APP client. In the identity chain, each service provider acts as a node. The client inputs the hash-processed user biological key HR_i and document information to the service node that has write access. Then, the service node writes the user's random key R_i to the IPFS, which

returns the hash address of the file upon successful deposit. The service node stores the user ID, hash of document information, and hash address in the blockchain. The service and public key will be returned to the client if the message is successfully uploaded.

When the user requests the service again, the authentication operation needs to be completed. The framework of the multistage authentication model based on blockchain technology is shown in Figure 7. The client packages the data firstly and sends the data to the server provider. In the first stage of authentication, the system retrieves the user's information in the blockchain based on the user's name. The strategy compares the user's document information and returns the hash address of the file if it matches. Successful authentication in the first stage indicates that the user's document information is verified. In the second stage of authentication, the user's random key HR_i is searched in the IPFS based on the preoutput of the first stage. This step compares whether the random key in the IPFS is consistent with the key generated at the time of input. If it is identical, it means the user's biometric information is authenticated successfully in the second stage.

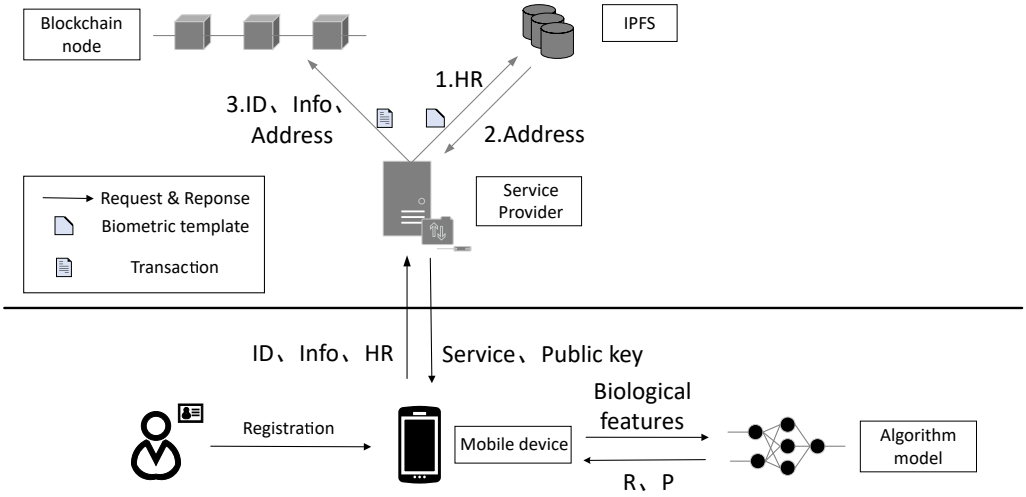


Figure 6. Registration model.

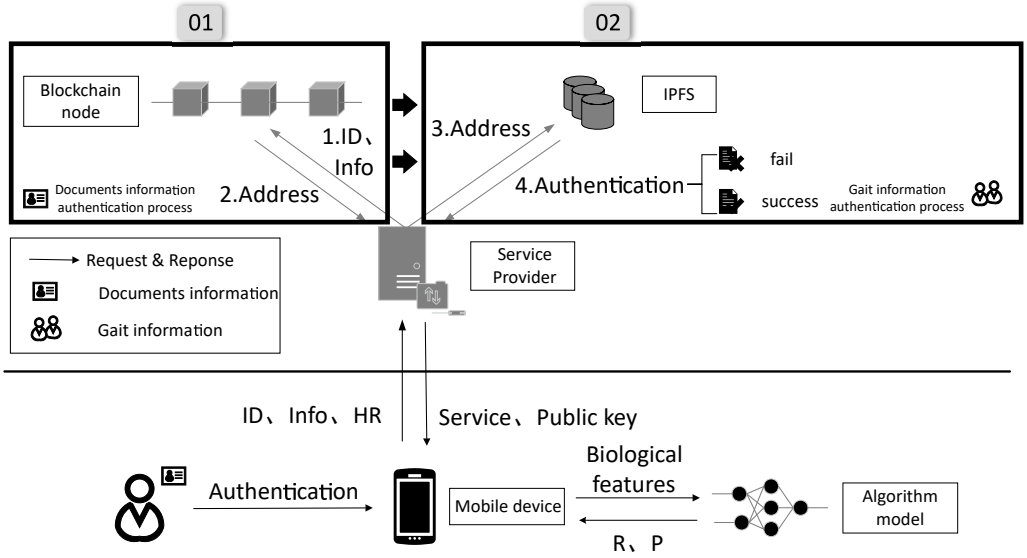


Figure 7. Authentication model.

4.2.1. Gait Key Extraction Scheme

The authentication process based on gait characteristics is shown in Figure 8. The black lines in the figure indicate the overall flow of the authentication model. Firstly, the gait data are collected through the built-in accelerometer and gyroscope sensors of the

smart devices. The gait data acquisition program is written by instruction code that can run in the background of devices. The data are recorded in real time when relevant activities are detected. The acquired data need to be preprocessed, including the removal of outliers, noise filtering, and gait cycle segmentation. Afterward, the processed gait feature data are extracted to generate a user's gait feature template. The feature template is stored in the database, and the authentication result can be judged according to the consistency of the template in the authentication stage.

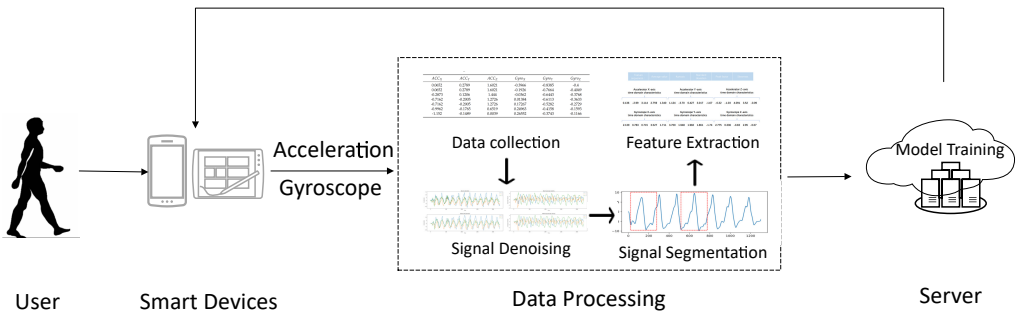


Figure 8. Gait recognition.

In this paper, body activity data were collected using a smartphone equipped with the Android operating system. The smartphone device has a triaxial accelerometer and a triaxial gyroscope. The acceleration sensor returns the three-axis acceleration applied to the device by the user, and it is measured in m/s^2. The gyroscope measures the rotation rate of the device, i.e., the rate of rotation of the device around each axis. It is measured in rad/s.

(1) Gait data collection

The dataset used in this paper was collected by a self-developed Android program. The data folder is automatically uploaded to the cloud after being collected. The dataset consists of 21 users. Each person collects 10 sample files. Each file contains the X axis, Y axis, and Z axis gait data of the accelerometer and gyroscope sensors. Table 3 shows the 3D acceleration and gyroscope data for one user. Notably, the gait signals collected by the proposed authentication scheme will not violate the privacy of the user. Our system will display a prompt message before the actual collection of gait information, and it will proceed to the next step only after obtaining the user's consent.

Table 3. The gait data for users.

ACC_X	ACC_Y	ACC_Z	$Gyro_X$	$Gyro_Y$	$Gyro_Z$
0.0652	0.2789	1.6021	−0.3966	−0.8385	−0.4
0.0652	0.2789	1.6021	−0.1926	−0.7664	−0.4049
−0.2873	0.1206	1.444	−0.0362	−0.6443	−0.3768
−0.7162	−0.2005	1.2726	0.01384	−0.6113	−0.3633
−0.7162	−0.2005	1.2726	0.17267	−0.5282	−0.2729
−0.9962	−0.1765	0.6519	0.24963	−0.4158	−0.1593
−1.152	−0.1489	0.0039	0.26552	−0.3743	−0.1166

We can simulate three-dimensional data from an experimenter's walking acceleration sensor and gyroscope sensor. Figure 9 is the waveform obtained for the X axis of the accelerometer. However, due to multiple factors such as different acquisition performance outcomes of the built-in sensors and unconscious jitter from the user's grip on the phone, the acquired signal is subject to noise fluctuations. There will be a large amount of anomalous data in the initial phase of the data collection. Therefore, we needed to preprocess the data.

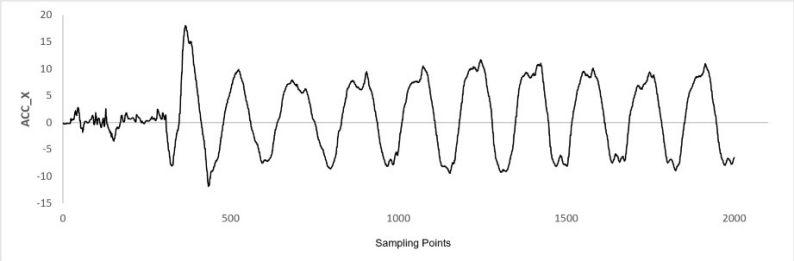


Figure 9. X axis waveform of the acceleration sensor.

(2) Data Preprocessing Processes

A. Signal denoising

The noise generated by built-in sensors is high-frequency random noise, while the gait signal belongs to a low communication signal. Walking frequency is usually below 20 Hz, and most of the energy of the signal is mainly concentrated in the frequency range of 1–5 Hz. Therefore, a Butterworth filter with maximum flatness was chosen in this paper to allow the low-frequency signal in the signal to pass through, while filtering or suppressing the high-frequency signal, thus retaining the appropriate step signal. The principle is as follows: the amplitude frequency mode-squared characteristic $H(u, v)$ of a Butterworth low-pass filter is given in Equation (2).

$$|H_a(j\Omega)|^2 = (\frac{1}{\sqrt{1+(\Omega/\Omega_c)^{2N}}})^2, \quad (2)$$

where N is the order of the filter, and Ω_c is the cut-off frequency of the low-pass filter. In this paper, the cut-off frequency of the 3rd-order Butterworth filter was set to 5 Hz. Figure 10 shows a comparison of the waveforms before and after denoising.

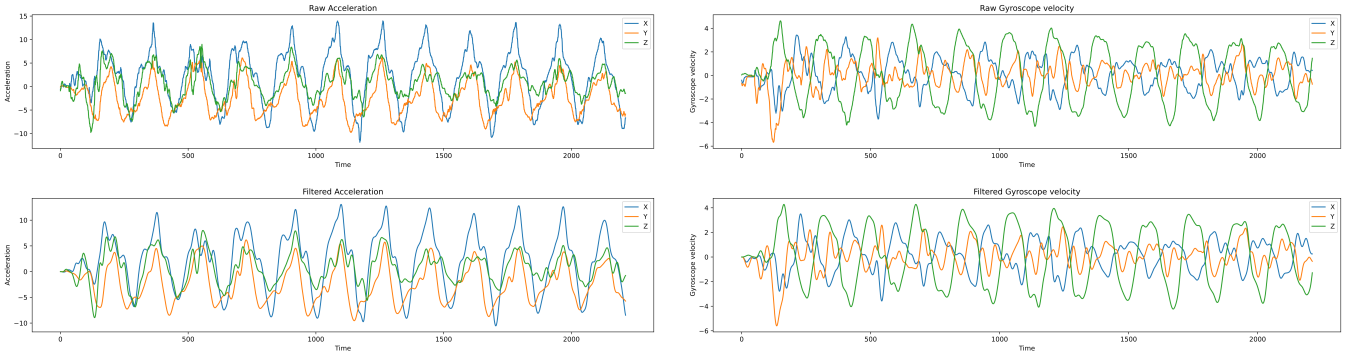


Figure 10. Comparison of gait signal before and after filtering.

B. Signal segmentation

The gait signal in the human body is periodic. The segmentation of the gait signal into sequence units facilitates the extraction of information about human gait characteristics. A gait cycle is defined as the time interval between two adjacent touches of the same foot to the ground. Using the gait cycle as a segmentation unit is a more feasible approach than dividing the gait signal based on a fixed length. It is also known that the acceleration signal reaches its maximum value when one foot touches the ground. Therefore, this paper used the maximum signal from the accelerometer as the starting point of the gait cycle.

The gait signals selected for this paper are an array of six channels, including the X, Y, and Z axis signals from the accelerometer and gyroscope sensors. As the experimenter holds the smart device in his hand and swings his arm naturally during data acquisition, his Z axis direction coincides with the direction of the user's acceleration. And the Z axis

direction is most sensitive to the perception of gait data. Therefore, the Z axis signal from the accelerometer sensor was used to complete the gait cycle detection in this paper.

As shown in Figure 11 below, this paper used a sliding window segmentation technique. The length of the human gait cycle sequence is in the range of 120–140, and there are slight differences between the individual gait signals collected in this paper. In order to ensure that the sliding time window contained the entire gait cycle, this paper used a sliding window of 150 in length to segment the collected gait data into a period. The cycles are indicated by red frames in the gait signal.

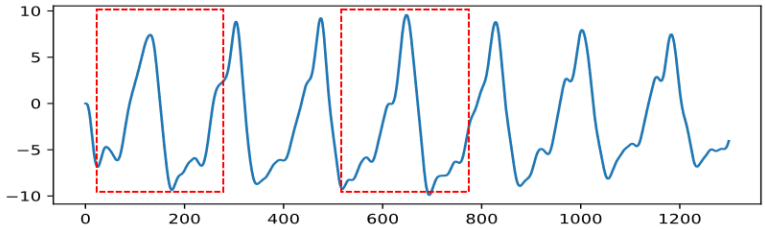


Figure 11. Sliding window Method.

(3) Feature point extraction

As the raw data from both the accelerometer and gyroscope sensors are high-dimensional, we needed to perform an effective feature extraction process for the data in this paper. In this paper, we extracted features from high-dimensional time series in the time domain. Common time domain features include maximum, minimum, mean, variance, and peak-to-peak. These time domain features all possess magnitudes and are therefore referred to as dimensioned eigenvalues. In addition, dimensionless features such as skewness, peak factor, and cliff factor can also be used. Appropriate feature selection not only improves the accuracy of authentication, but also increases the efficiency of identification and speeds up the process of identification and authentication. Therefore, in this paper, the five features of mean, kurtosis, standard deviation, peak factor, and skewness were finally selected as the basis for authentication. The sample format of the gait data feature extraction is shown in Figure 12, and the 30-dimensional data features will be used in the subsequent authentication.

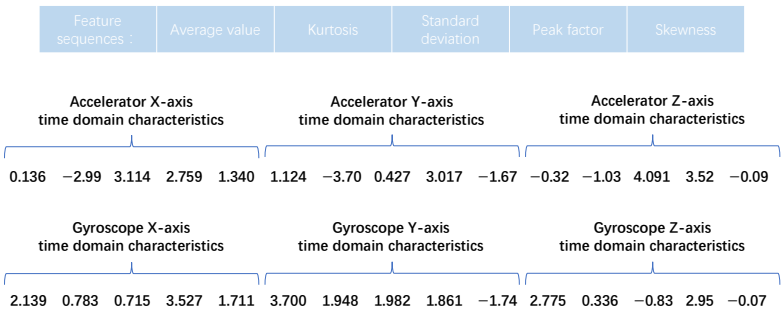


Figure 12. Sample format of data extracted from gait features.

(4) Gait secret key

According to the fuzzy extractor and behavioral feature extraction scheme introduced above, the key extraction process based on the gait is shown in Figure 13. In the registration phase, the user performs gait data collection and feature extraction. Then, we input the extracted features ω_i into the fuzzy extractor and generate a random key R and a public string P according to the Gen algorithm. The random key R is hashed and stored in the IPFS as the gait key, while the public information P is stored on the chain. When the user is authenticated, the user's gait characteristics ω_i are collected again, and P is extracted from

the blockchain. If the biometric error of the two inputs is satisfied within a given range, i.e., $dis(\omega_i, \omega_i{}') \leq \epsilon$, then the random key R' can be recovered from $\omega_i{}'$ and P. We hash the random key R' and extract the original hash (R) stored in the IPFS. If $hash(R') = hash(R))$ is satisfied, then the biometric verification is successful.

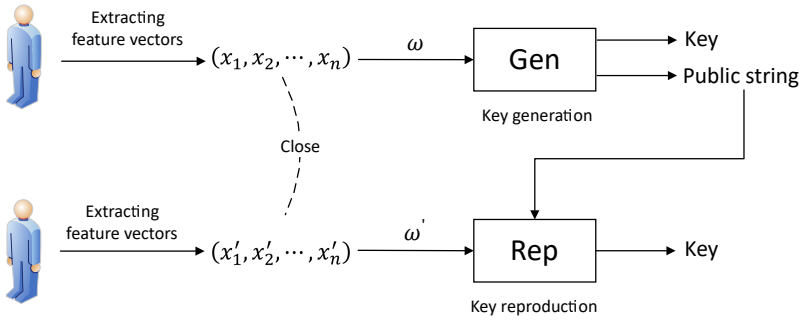


Figure 13. Key extraction process based on gait.

4.2.2. IdentiChain

The types of blockchain are divided into public blockchain, consortium blockchain, and private blockchain. In the public blockchain represented by Bitcoin, all nodes can be accessed at any time to read data, download the complete blockchain ledger, and send confirmed transactions. Users can compete for mining rewards by engaging in mining, which essentially involves solving puzzles by expending arithmetic power to compete for bookkeeping rights. As a result, users are rewarded with virtual tokens by the system. Although this consensus model ensures maximum decentralization, it is not only wasteful, but also inefficient. It is obviously not suitable for the immediate response needed in this paper. Moreover, there is no need to use virtual tokens in this system, so it is not possible to reward miners with tokens for mining. Fabric functions as a federated blockchain, thereby providing access to nodes through a public key infrastructure (PKI), which allows authorized nodes to participate in the federated chain network. Members participating in the blockchain network need to have previously registered with the blockchain to obtain a registration certificate that can be generated based on the type of entity. Fabric identifies users by issuing digital certificates, each of which corresponds to its membership service provider number on a one-to-one basis. In addition, Fabric uses a gRPC-based gossip protocol to ensure consistent messaging between nodes. The gossip protocol involves a randomly selected node that broadcasts messages and thus can improve network load performance and reduce the likelihood of malicious attacks. The greatest strength of Hyperledger Fabric is its pluggable modules such as consensus mechanisms, rights management, and ledger mechanisms. It is designed to optimize privacy and performance through privileged access, thus enabling enterprises to limit the mobility and visibility of information to legitimate users.

In Fabric blockchain networks, nodes are the main communication bodies. Multiple nodes of different types can be deployed on different servers to form a blockchain network, or they can run on the same physical server by using docker containers. In Hyperledger Fabric, there are different types of nodes: client nodes, peer nodes, orderer nodes, and CA nodes. The Fabric network architecture used in this paper is shown in Figure 14.

Smart contracts in the Hyperledger Fabric are called chain codes, and their main function is to execute transactions and provide access to data on the chain. The chain code is the only way to interact with the blockchain storage, and all business logic implementation needs to be done through the chain code. During the development of the chain code, the init initialization function and the invoke call function of the chain code interface need to be implemented first. The init function is used to initialize the chain code parameters. The invoke function usually contains several functions, such as a query function for the data on the chain and a function to modify the data on the chain. In order to reduce development

costs and ease of use, this article has been written in Node.js for chain code and developed using the VS Code editor.

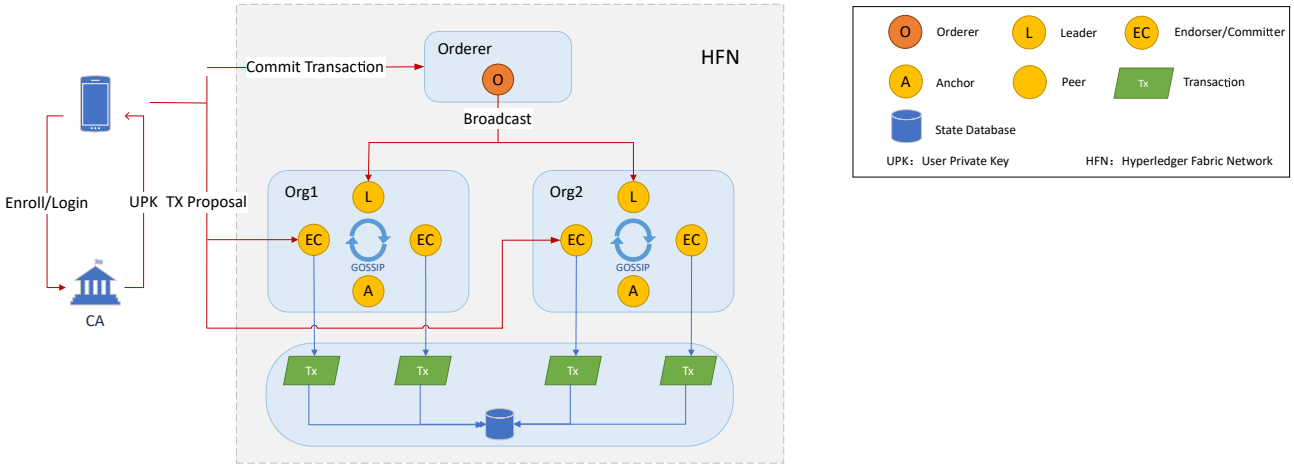


Figure 14. The Fabric network architecture.

4.3. System Architecture

This section introduces a continuous authentication system based on blockchain. This system provides a distributed mechanism for identity authentication, rather than relying on a central server to handle authentication operations. This decentralized mechanism eliminates the problem of single-point failures and provides a reliable authentication solution for an identity system. In this scheme, the blockchain network is considered as a federated chain where only authenticated nodes can participate in the operation. The core operations of the system such as template management, identity registration, and identity verification are implemented by smart contracts in the blockchain.

The registration and authentication process of the system is as follows.

4.3.1. Registration

Any service node uses this system to authenticate users. When a user wishes to apply for the services, the user needs to perform a registration operation and become a legitimate user. The user generates the user's public and private keys using elliptic curve cryptography (ECC). The registration process is shown in Figure 15. The specific steps for registration are as follows:

Step 1: Users U_i upload identification ID_i, certificate information($Info$) to the client, and obtain gait characteristics information through the sensor device.

Step 2: The gait recognition model extracts a template ω_i of the user's gait features. The fuzzy extractor is used to fuzzy extract ω_i to obtain the public information P_i and the random key R_i, and then it hashes the random key R_i to obtain the HR_i. The client locally stores the user's public information P_i.

Step 3: The client generates the public key EPK and the private key ESK according to the elliptic curve algorithm. Then, the client sends the user ID_i, HR_i, EPK timestamp T_0 and $hashInfo$ to the service provider.

Step 4: The service provider writes the HR_i to the IPFS. And the IPFS returns the hash address of the file.

Step 5: The endorser verifies and simulates the requests sent by the server provider. The server sends the transaction to the orderer when it receives enough backing support for the endorsement.

Step 6: The committer is responsible for maintaining the blockchain and performing final checks on the transactions. The transactions are written to the ledger after the check is passed.

Step 7: The service provider completes the registration of the user and returns the public key PK and the service.

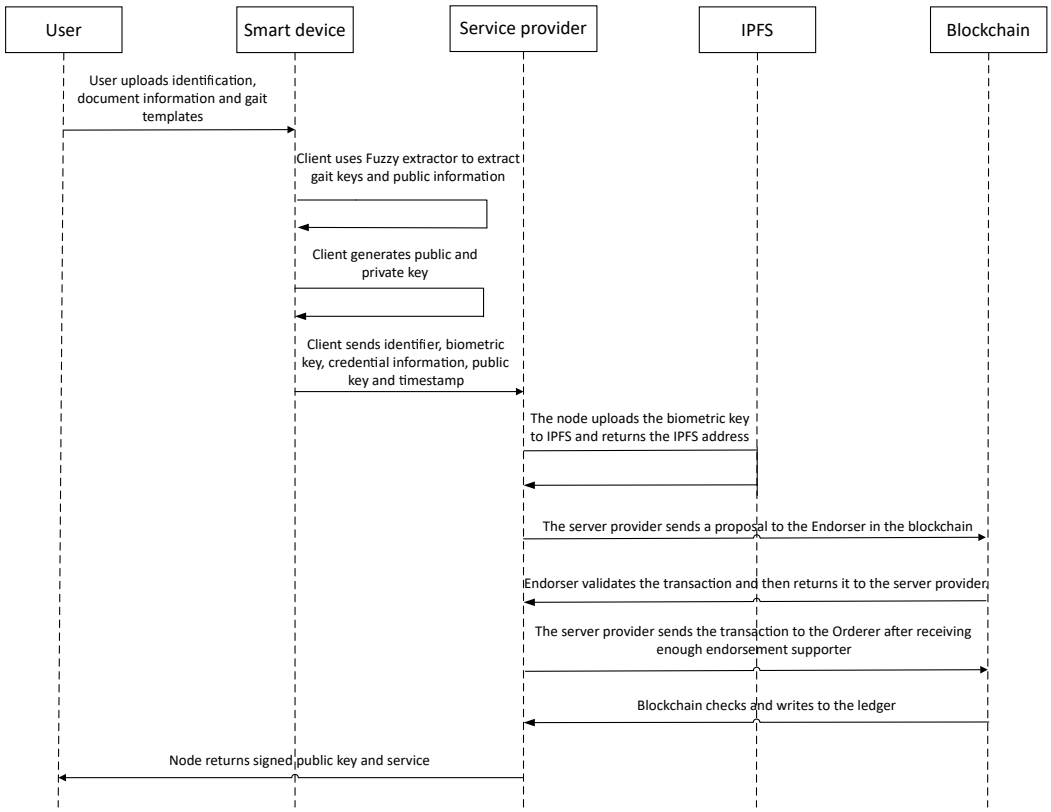


Figure 15. Process of user registration.

4.3.2. Authentication

When a legitimate user wishes to access the services of a service provider, then the authentication process needs to be completed. The authentication process is shown in Figure 16. The specific steps for authentication are as follows:

Step 1: The user enters the ID_i, biometric information, and $Info$ into the client.

Step 2: The client extracts current login time T_1 and sends the $A = PK(hashInfo', T_1)$ to the service provider. Then, the service provider decrypts A with a service-side private key SK and obtains $hashInfo'$, T_1.

Step 3: The service provider sends $B = EPK(hashInfo', T_1, T_0)$ to the client. The client decrypts B with its own private key ESK and obtains $hashInfo'$, T_1, and T_0. The client compares the results with T_1 saved in step 2; if they are equal, the service side verification is completed. At the same time, users can check if the T_0 moment matches the registered login time to avoid someone from stealing it.

Step 4: The client extracts the user's biometric template $\omega_i{}'$ and recovers the user's random key $R_i{}'$ based on $\omega_i{}'$ and the locally stored public information P. Then, the random key is hashed to generate $HR_i{}'$. Then, the client inputs ID_i, HR_i', and $C = PK(hashInfo')$ to the service provider.

Step 5: The service provider retrieves the user's index ID_i in the blockchain and decrypts C using the private key SK to obtain $hashInfo'$. Then, the decrypted information is compared with the $hashInfo$ entered for registration. If the comparison is successful, the system will return the file hash address in the IPFS. The user's credential information is verified.

Step 6: The service provider downloads the user's HP_i in the IPFS system. If the HR_i is the same as the $HR_i{}'$, the user's biometric authentication succeeds.

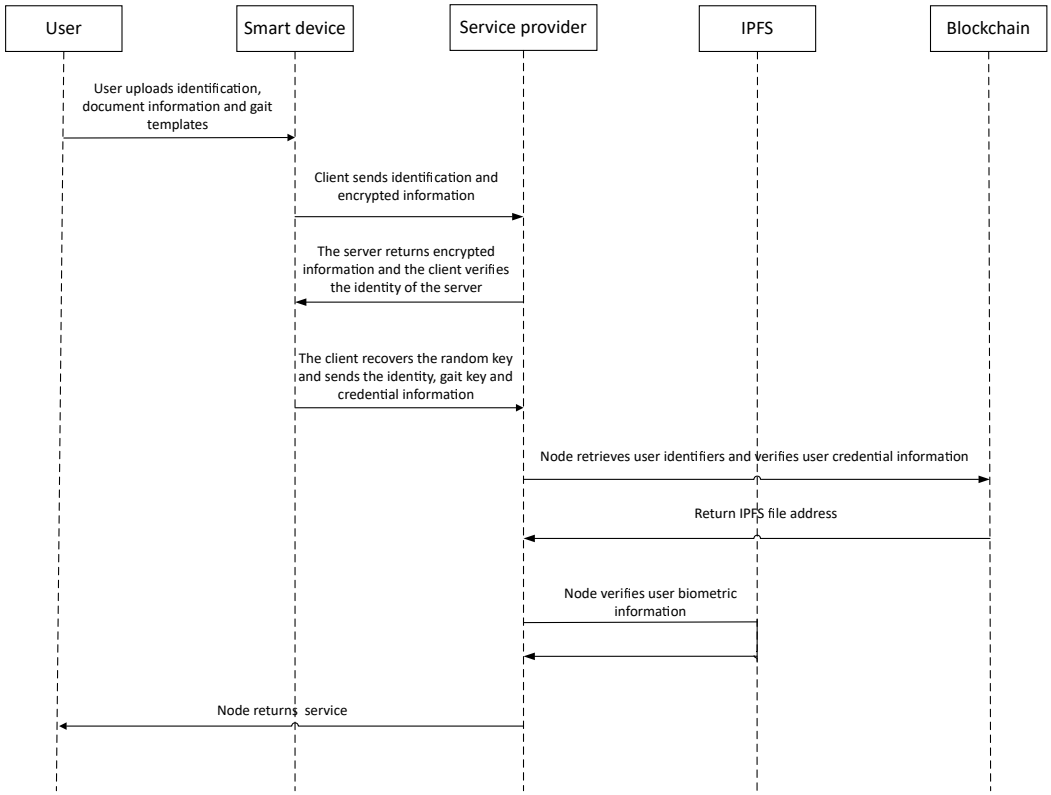


Figure 16. Process of user authentication.

5. Experimental Simulation

5.1. System Operating Environment

In order to verify the feasibility of the solution, we conducted a simulated experiment using Python 3.8. We used the Ubuntu 22.04 operating system with an Intel Core i5-12500H CPU and a 16 GB RAM system. The system used the Python and Go languages to complete the relevant implementations in the experiment.

This section describes the operational environment of the continuous authentication system, including the gait recognition algorithm, fuzzy extractor, blockchain part and IPFS. The gait features were extracted from the time and frequency domains. Feature selection was carried out using principal component analysis to obtain the features with the best combination. The fuzzy extractor was written based on Python. We built a test environment based on Hyperledger Fabric 2.4, which was deployed in solo mode. The data storage component uses the IPFS distributed storage and a CouchDB database.

5.2. Implementation of IPFS

The interaction with the IPFS can be done directly from the command line or through an API interface, with the Go language backend eventually completing the interaction with the IPFS. In this article, we chose to use the command line for interactive testing to implement the IPFS distributed file storage system environment.

5.3. Implementation of Blockchain Networks

(1) Environment settings

The blockchain was implemented using the Hyperledger Fabric 2.4 technology framework. Smart contracts were developed using the Go language. The nodes of the blockchain network were run in Docker containers, which were configured according to the image via Docker compose files. This included one orderer node, two Org organizations, one CouchDB service, and one CA service, and each organization included four peer nodes; each peer node was configured with the appropriate role.

(2) Consensus mechanism

We used the Solo single-node consensus algorithm in this paper. After a node in a federation chain submits a transaction request, the orderer node sorts and packages the transactions that meet the conditions, and it backs up the transactions once at each node.

(3) Smart contract

The system designed two smart contracts for registering information and performing user authentication. The parameters to be stored in the smart contracts and the corresponding functions needed to be determined. We stored the following data in the smart contracts: (1) user name index, (2) timestamp of each transaction, (3) signInfo hash of the signed credential information, and (4) public information P. The details are as follows.

In the registration information smart contract, the algorithm is defined as follows (Algorithm 1):

Algorithm 1: Smart Contract I : Identity Registration

Input : Identifier ID_i, Document information $Info$, Feature Key HR_i, Client public key EPK
Use identifier ID_i as key to find its corresponding value
if there is no corresponding value, then
{ Set the username to the value of identifier ID_i. Save HR_i as a feature key into IPFS and deposit $Info, EPK$ into the blockchain.
return success
}
else
{if $Info' = Info$ then
    {Update feature key HR_i value into IPFS and save Info, EPK into blockchain
    return true
    }
    else return false
{

In smart contract I, the user's identity information is first verified whether it exists; if not, the user's identity is directly saved. If it exists, the ID information of the already existing identity information user is first verified. If it matches with this caller, the user's identity information is updated. Otherwise, it returns an error.

The authentication smart contract algorithm is defined as follows (Algorithm 2):

Algorithm 2: Smart Contract II: Identity Verification

Input : Identifier ID_i, Document information $Info'$, Feature Key HR_i', Login time T_0
Get identifier ID_i from the input information and use ID_i as key to get the corresponding value
if there is no corresponding value, then
    return false
else
{ Decrypt the message using the private key to get $Info'$
    if $Info' = Info$ then
        return IPFS address
            { if $HR_i' = HR_i$ then
            return true
            }
    else return false

In the authentication smart contract, the implementation looks up the identity information based on the user's name. If the information exists, the signature is authenticated, and the IPFS address is returned for biometric authentication. Then, if the biometric authentication is successful, it means the user's identity is successfully authenticated.

(4) Access control

We performed the on-chain read step after completing the chain code publishing and deploying the smart contract. It includes adding the user's identity information, querying the fabric network for the user's identity information, using the get operation and the user updating biometric information. Figure 17 below shows the successful authentication of the user's identity.



Figure 17. User authentication process.

6. Scheme Analysis

6.1. Security Evaluation

In this paper, the security of the system was analyzed separately using nonformal and formal methods, which are described in the following.

6.1.1. Security Analysis

Decentralization: There is no longer a reliance on a single central authority in the structure of the blockchain. As a result, there will be no single point of failure of the central server that could bring down the blockchain network system. At the same time, the distributed structure prevents the central server from monopolizing and tampering with the data. This feature can protect the legal rights of users effectively.

User Anonymity: In order to ensure the security of the user's identity information, the identity of the user in this scheme is transmitted as the result of a hash operation. The blockchain credentials store the user's signed identity information $signInfo$. The unidirectional nature of the hash function makes it computationally challenging to deduce the user's identity plaintext $Info$ by backtracking $signInfo$. In addition, the encrypted public information $Hash(P_i)$ is stored in the IPFS. Even if a malicious person obtains the encrypted message, it cannot be decrypted in reverse. The public message P_i does not leak too much information about the random string R_i. As a result, this solution can prevent the leakage of user identity information to a certain extent and ensure the privacy and security of the identities of the users on the chain.

Conditional Traceability and Nonrepudiation Property: Due to the traceability feature of the blockchain, the communication information between the user and the server is permanently recorded in the blockchain in the form of blocks. This means all data can be traced. And, given that all of the user's transactions are digitally signed with a private key known only to the server, the server cannot deny the act of signing the trust credentials; thus, the user cannot deny the interaction with the server.

Resistance to Other Attacks: In common forgery attacks, the attacker publishes a malicious message by forging the identity of the signatory. In this scheme, it is very difficult for an attacker to forge an identity. It is assumed that the attacker intercepts the authentication request message sent by the user to the service node. However, it is not possible to decrypt the request message, because the private key of the service node is not available. Even if the private key of the service node is leaked, only the server side has the corresponding key pair and writes the public key to the blockchain ledger. Consequently, the attacker is unable to obtain a legitimate digitally signed message. This shows that the scheme is resistant to man-in-the-middle (MitM) attacks.

6.1.2. Logic Proof by BAN Logic

Formal analysis is an analysis tool commonly used in the design phase of security authentication protocols. The purpose of the analysis is to authenticate the legal identity of the subjects participating in the protocol. The BAN logic analysis of authentication protocols consists of the following four steps:

1. Protocol idealization: This involves deriving idealized protocols from real agreements. It means that the messages in the protocol are represented in the corresponding BAN logic language;
2. Initial state assumptions: This step identifies the initial belief assumptions and security objectives of the protocol and represents them in logical notation;
3. Logical Reasoning Process: This involves reasoning about messages based on initial assumptions and logical reasoning rules to arrive at the final beliefs of the individual certified entities;
4. Conclusion: The conclusion of the reasoning is analyzed. If the conclusion achieves the purpose for which the protocol was designed, then the protocol is secure; otherwise, the protocol has security flaws.

The BAN logic consists of three basic terms: the proposition $\{X\}$, subject $\{P, Q\}$, and encryption key K. The notation used in the analysis of BAN logic is shown in Table 4.

Table 4. The BAN logic notations.

Notation	Description
$P \mid\equiv X$	P believes in the authenticity of statement X.
$\#(X)$	Statement X is fresh and has not been sent before.
$P \mid\Rightarrow X$	P has jurisdiction over declaration X.
$P \mid\sim X$	P once said X.
$P \triangleleft X$	P obtains statement X.
$\{X\}_K$	Encryption of statement X with key K.
$P \overset{K}{\longleftrightarrow} Q$	Communication between P and Q with shared key K.

The logical inference rules of the BAN formalize the inference of the authentication protocol. The inference rules specify that Y is valid if X_1, X_2, $\cdots$, and X_n all stand. The following logical rules are considered in this paper:

(1) Message-Meaning Rule (MMR): This rule deals with the interpretation of messages. This formula indicates that P believes the key K shared between it and Q and receives a message X encrypted with K. Then, P believes that it was Q that sent the belief X.

$$Message\text{-}Meaning\ Rule\ (MMR): \frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X} \tag{3}$$

(2) Nonce Verification Rule (NVR): This rule defines how to assert that a message is fresh and that the sender believes the message is fresh. The formula expresses that P believes that X was sent by Q, and P believes that X is fresh. Then, P believes that Q sent X over the course of the protocol.

$$Nonce\ Verification\ Rule\ (NVR): \frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\sim X} \tag{4}$$

(3) Jurisdiction Rule (JR): This rule states that P believes Q has jurisdiction. The formula indicates that P believes X if P believes Q has jurisdiction over X and trusts Q to believe X.

$$Jurisdiction\ Rule\ (JR): \frac{P \mid\equiv (Q \mid\Rightarrow X), P \mid\equiv (Q \mid\equiv X)}{P \mid\equiv X} \tag{5}$$

(4) Freshness Propagation Rule (FPR): This rule states that if one element of a formula is fresh, then the whole formula is deemed to be fresh.

$$Freshness\ Propagation\ Rule\ (FPR): \frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)} \tag{6}$$

(5) Belief Rule (BR): Belief rules refer to consistency in the different operations of the cascading and splitting of messages, as well as to the transferability of the beliefs in such operations.

$$Belief\ Rule\ (BR): \frac{P \mid\equiv (X,Y)}{P \mid\equiv (X)} \tag{7}$$

We use S_i for the server and U_i for the user. Based on the proposal description in this paper, we have set the following assumptions in the scheme:

$A1 : S_i \mid\equiv PK$

$A2 : U_i \mid\equiv EPK$

$A3 : S_i \mid\equiv \#(T_0)$

$A4 : U_i \mid\equiv \#(T_1)$

$A5 : S_i \mid\equiv U_i \mid\Rightarrow U_i \overset{PK}{\longleftrightarrow} S_i$

$A6 : U_i \mid\equiv S_i \mid\Rightarrow S_i \overset{EPK}{\longleftrightarrow} U_i$

The procedure of the identification protocol steps in this paper can be described formally in the BAN logic language, where U_i stands for the client and S_i for the server side. In this process, messages transmitted in plaintext are irrelevant for the analysis of the arguments that can be removed. Therefore, the idealized language description of this scheme is as follows:

$Message1 : U_i \rightarrow S_i : PK(ID_i, hashInfo, T_0, U_i \overset{PK}{\longleftrightarrow} S_i)$

$Message2 : S_i \rightarrow U_i : EPK(ID_i, hashInfo', T_0, T_1, S_i \overset{EPK}{\longleftrightarrow} U_i)$

$Message3 : U_i \rightarrow S_i : (hashInfo', ID_i, BR_i)$

In addition, the following security objectives are identified in this paper to demonstrate our scheme:

$Goal1.\ S_i \mid\equiv S_i \overset{K}{\longleftrightarrow} U_i$

$Goal2.\ U_i \mid\equiv S_i \overset{K}{\longleftrightarrow} U_i$

$Goal3.\ S_i \mid\equiv U_i \equiv S_i \overset{K}{\longleftrightarrow} U_i$

$Goal4.\ U_i \mid\equiv S_i \equiv S_i \overset{K}{\longleftrightarrow} U_i$

The BAN logical proof is carried out using the above logical rules, which are idealized forms and assumptions to prove given goals. This agreement is evidenced by the following process:

- According to the $Message1$, we can obtain E_1:
$$E_1 : S_i \triangleleft \left\{ ID_i, hashInfo, T_0, U_i \overset{PK}{\longleftrightarrow} S_i \right\}$$
- We employ MMR using E_1 and assumption A_1 to obtain E_2:
$$E_2 : S_i \mid\equiv U_i \mid\sim \left\{ ID_i, hashInfo, T_0, U_i \overset{PK}{\longleftrightarrow} S_i \right\}$$
- We employ the FPR using E_2 and assumption $A3$ to obtain E_3:

$$E_3 : S_i \mid\equiv \#\left\{ID_i, hashInfo, T_0, U_i \overset{PK}{\longleftrightarrow} S_i\right\}$$

- We employ the NVR using E_2 and E_3 to obtain E_4:

$$E_4 : S_i \mid\equiv U_i \mid\equiv \left\{ID_i, hashInfo, T_0, U_i \overset{PK}{\longleftrightarrow} S_i\right\}$$

- We employ the BR using E_4 to obtain E_5:

$$E_5 : S_i \mid\equiv U_i \mid\equiv U_i \overset{PK}{\longleftrightarrow} S_i$$

- In addition, we use JR and assumption $A5$ using E_5, which gives us the following:

$$E_6 : S_i \mid\equiv U_i \overset{PK}{\longleftrightarrow} S_i$$

- According to the $Message2$, we can obtain E_7:

$$E_7 : U_i \triangleleft \left\{ID_i, hashInfo', T_0, T_1, S_i \overset{EPK}{\longleftrightarrow} U_i\right\}$$

- We employ the MMR and assumption A_2 to obtain E_8:

$$E_8 : U_i \mid\equiv S_i \mid\sim \left\{ID_i, hashInfo', T_0, T_1, S_i \overset{EPK}{\longleftrightarrow} U_i\right\}$$

- We employ the FPR using E_8 and assumption $A2$ to obtain E_9:

$$E_9 : U_i \mid\equiv \#\left\{ID_i, hashInfo', T_0, T_1, S_i \overset{EPK}{\longleftrightarrow} U_i\right\}$$

- We employ the NVR using E_8 and E_9 to obtain E_{10}:

$$E_{10} : U_i \mid\equiv S_i \mid\equiv \left\{ID_i, hashInfo', T_0, T_1, S_i \overset{EPK}{\longleftrightarrow} U_i\right\}$$

- We employ the BR using E_{10} to obtain E_{11}:

$$E_{11} : U_i \mid\equiv S_i \mid\equiv S_i \overset{EPK}{\longleftrightarrow} U_i$$

- In addition, we use JR and assumption $A6$ using E_{11}, which gives us the following:

$$E_{12} : U_i \mid\equiv S_i \overset{EPK}{\longleftrightarrow} U_i$$

After employing the BAN logical formal analysis, it is evident that the protocol is able to achieve the desired security goals. Therefore, the protocol in this paper is secure.

6.2. Performance Analysis

In this section, we analyzed the performance of this scheme on IoTs devices in terms of both computational and communication overhead, and we compared it with existing IoTs device authentication schemes.

6.2.1. Functionality Comparison

Table 5 compares the functional characteristics of this paper's scheme with those of related user authentication schemes. The schemes of Omar [47] and He et al. [48] are vulnerable to biometric misidentification and offline password-guessing attacks. The Toutara et al. [49] scheme, on the other hand, is not resistant to replay and man-in-the-middle attacks. The Das et al. [50] scheme cannot avoid the problem of a single point of failure. We can note that only our solution is able to meet the known security requirements in the core network domain and provides a federated blockchain.

Table 5. Comparison of functional features.

Schemes	Omar [47]	Toutara [49]	Das [50]	He [48]	Proposed
User anonymity	Yes	Yes	Yes	Yes	Yes
Nonrepudiation property	Yes	Yes	Yes	Yes	Yes
Avoidance of single-point failure	Yes	Yes	No	Yes	Yes
Tamper proof	Yes	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes
Mutual Authentication	Yes	No	Yes	Yes	Yes
Replay attack	Yes	No	Yes	Yes	Yes
Man-in-the middle attack	Yes	No	Yes	Yes	Yes
Resists biometric recognition error	No	Yes	Yes	No	Yes
Offline password-guessing attack	No	Yes	No	No	Yes

6.2.2. Performance Testing

The performance of the blockchain determines the authentication efficiency of the system. Therefore, we conducted further experimental tests on the throughput of the blockchain network in this paper. This system performance test was conducted mainly using the Hyperledger Caliper performance benchmarking framework tool. Caliper is able to test metrics such as throughput, CPU usage, and the latency of smart contracts in the built blockchain network in a user-defined manner. In this article, we analyzed and compared the data based on the six items in the report. Among them, the query operation includes functions such as querying user credential information. The update operation includes functions such as updating behavioral characteristics and updating credential information. As shown in Table 6, this paper tested the QueryRecordID, QueryCreditInfo, AddRecordID, AddCreditInfo, and AddCreditvalue.

Table 6. Test chain code functions and test parameters.

Test ChainCode	Number of Sent Transactions	Transmission Rate
Qurey ChainCode	3000	100, 150, 200, 250, 300, 350, 400
Update ChainCode	3000	50, 75, 100, 125, 150, 175, 200

In testing the query chain code function, we fixed the number of transactions to be sent 3000 times and changed the sending rate to 100 TPS, 150 TPS, 200 TPS, 250 TPS, 300 TPS, 350 TPS, and 400 TPS. We tested the throughput and average latency of the corresponding system. We performed seven sets of tests on the system. In the tests for the query operation, the sending frequency increased steadily until it reached 300 TPS, and the average latency was maintained within 0.2 s. The throughput of the system reached a maximum of 326 TPS when the sending rate was 300 TPS. As is shown in Figure 18, at maximum transaction processing, our model could achieve a high throughput of 326 TPS, which is 8.6% [51] higher compared to the 300 TPS throughput achieved by the proposed model in the related work. After the sending frequency exceeded 300 TPS, the throughput started to decrease, and the average latency increased rapidly. This is due to the fact that at send rates greater than 300 TPS, the limit of how many transactions the system can handle is exceeded: transactions have started to block, and the TPS cannot continue to increase.

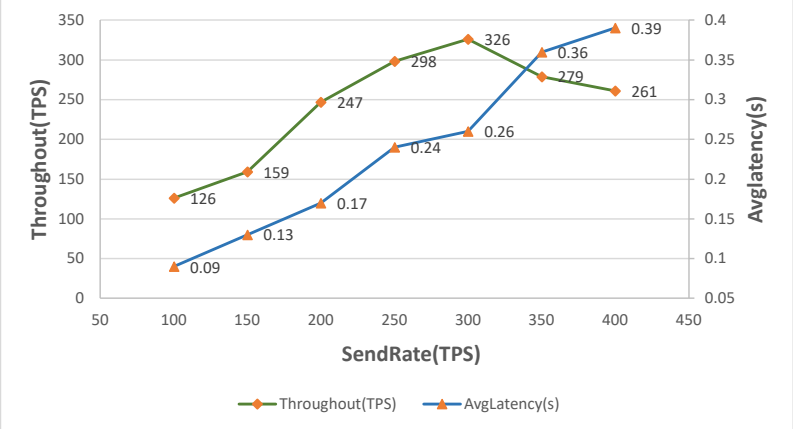


Figure 18. Query chain code test result.

As shown in Figure 19, the same method was used in this paper to perform seven sets of tests on the update chain code using the test update chain code function. Among them, the throughput of the system reached the highest when the sending rate was 150 TPS. In the subsequent sets of tests, the throughput of the system gradually decreased, and the average delay increased as the processing capacity of the Fabric nodes decreased.

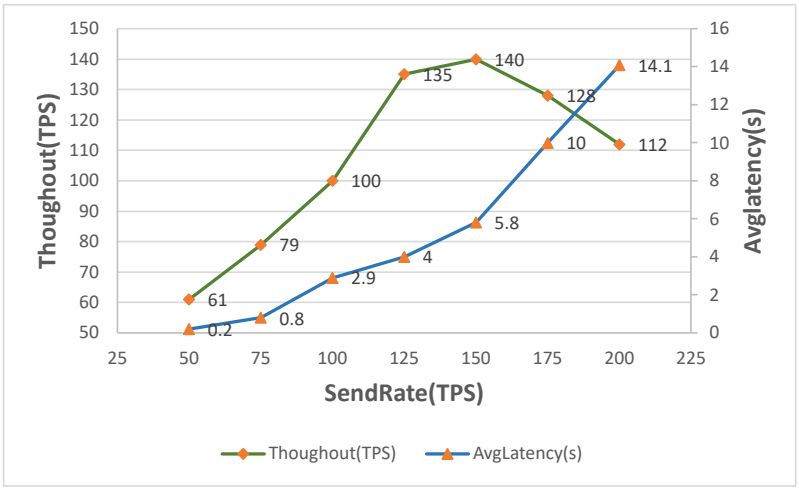


Figure 19. Update chain code test result.

In addition, Caliper also reflects the occupancy of resources according to individual nodes. The query performance test in this paper focused on the performance of the peer and orderer nodes, as well as other components to query data. The average is taken based on the occupancy before the maximum throughput TPS is reached. As shown in Table 7, the average CPU usage and average memory usage of the peer nodes and orderer nodes were obtained.

Table 7. Resource utilization table.

Type	Node	Average Memory Consumption (MB)	Average CPU Utilization (%)
Docker	orderer	68.3	5.43
Docker	peer0.org1	309.8	12.6
Docker	peer1.org1	314.2	12.97
Docker	peer0.org2	336.1	14.07
Docker	peer1.org2	321.4	13.85

7. Conclusions

Network identity management is an important part of the security of intelligent connected vehicles (ICVs). Identity authentication technology is an important guarantee for maintaining Internet of Vehicles security and is the first barrier to network identity management. In this paper, we proposed an authentication scheme based on Fabric network architecture by combining authentication with the blockchain. The proposed model employs a combination of the blockchain and the IPFS to store user information, which avoids the single-node failure problem of traditional authentication. In addition, we use a fuzzy extractor to convert the user's biometrics information into ciphertext form. In this way, even in the case of database leakage, the attacker cannot present the user's real biometrics based on the obtained ciphertext public information. The results show that our system has high security against mainstream IoTs attacks and improves the efficiency of authentication. Compared to the related research models, the throughput of the query chain code was improved by 8.6%.

Our work confirms the promising applications of combining the blockchain with identity authentication. To the best of our knowledge, this is the first work that applies the blockchain to multistage continuous authentication. However, the existing blockchain technology suffers from capacity constraints, which will directly affect the rate of response for authentication. If it needs to be put into use on a large scale, we need to improve smart contracts and consensus algorithms to further increase the authentication efficiency. In our future work, we will consider the practical realization of the proposed system in IoT

devices. This would allow us to identify potential weaknesses to improve the scalability of the current proposal.

Author Contributions: Y.Y. performed the data gathering and preprocessing, contributed to the construction of the environment, executed the detailed analysis, and wrote some sections. X.Z. set the objectives of the research, contributed to the design of the algorithms, guided the data acquisition, validated the data processing and results, designed the paper, wrote some sections, and performed the final corrections. H.H., H.L., R.H. and Z.W. contributed to the design of the algorithms and the data acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 61802252.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available from the author upon request. The data are not publicly available due to the data was collected individually by laboratory personnel.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, J.; Letaief, K.B. Mobile edge intelligence and computing for the internet of vehicles. *Proc. IEEE* **2019**, *108*, 246–261. [CrossRef]
2. Joo Fong, T.; Abdullah, A.; Jhanjhi, N.Z.; Supramaniam, M. The coin passcode: A shoulder-surfing proof graphical password authentication model for mobile devices. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 302–308.
3. Hemalatha, S. A systematic review on Fingerprint based Biometric Authentication System. In Proceedings of the IEEE 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; pp. 1–4.
4. Guerar, M.; Migliardi, M.; Palmieri, F.; Verderame, L.; Merlo, A. Securing PIN-based authentication in smartwatches with just two gestures. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5549. [CrossRef]
5. Patel, V.M.; Chellappa, R.; Chandra, D.; Barbello, B. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Process. Mag.* **2016**, *33*, 49–61. [CrossRef]
6. Li, W.; Lu, W.; Sha, X.; Xing, H.; Lou, J.; Sun, H.; Zhao, Y. Wearable gait recognition systems based on MEMS pressure and inertial sensors: A review. *IEEE Sens. J.* **2021**, *22*, 1092–1104. [CrossRef]
7. Shahidinejad, A.; Ghobaei-Arani, M.; Souri, A.; Shojafar, M.; Kumari, S. Light-edge: A lightweight authentication protocol for IoT devices in an edge-cloud environment. *IEEE Consum. Electron. Mag.* **2021**, *11*, 57–63. [CrossRef]
8. Irshad, A.; Usman, M.; Chaudhry, S.A.; Naqvi, H.; Shafiq, M. A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4425–4435. [CrossRef]
9. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.
10. Reese, K.; Smith, T.; Dutson, J.; Armknecht, J.; Cameron, J.; Seamons, K. A usability study of five two-factor authentication methods. In Proceedings of the Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019), Santa Clara, CA, USA, 12–13 August 2019; pp. 357–370.
11. Mwitende, G.; Ye, Y.; Ali, I.; Li, F. Certificateless authenticated key agreement for blockchain-based WBANs. *J. Syst. Archit.* **2020**, *110*, 101777. [CrossRef]
12. Kamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Decentralized Business Review; US Sentencing Commission: Washington, DC, USA, 2008.
13. Zhang, G.-G.;Wan Z. Blockchain-based IoT security authentication system. In Proceedings of the 2021 International Conference on Computer, Blockchain and Financial Development (CBFD), Nanjing, China, 23–25 April 2021; pp. 415–418.
14. Huckle, S.; Bhattacharya, R.; White, M.; Beloff, N. Internet of things, blockchain and shared economy applications. *Procedia Comput. Sci.* **2016**, *98*, 461–466. [CrossRef]
15. Khan, U.; An Z.Y.; Imran, A. A blockchain ethereum technology-enabled digital content: Development of trading and sharing economy data. *IEEE Access* **2020**, *8*, 217045–217056. [CrossRef]
16. Gupta, Y.; Shorey, R.; Kulkarni, D.; Tew, J. The applicability of blockchain in the Internet of Things. In Proceedings of the 2018 10th International Conference on Communication Systems and Networks (COMSNETS), Bengaluru, India, 3–7 January 2018; pp. 561–564.
17. Gai, K.; Guo, J.; Zhu, L.; Yu, S. Blockchain meets cloud computing: A survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2009–2030. [CrossRef]

18. Javaid, M.; Haleem, A.; Singh, R.P.; Khan, S.; Suman, R. Blockchain technology applications for Industry 4.0: A literature-based review. *Blockchain Res. Appl.* **2021**, *2*, 100027. [CrossRef]
19. Berdik, D.; Otoum, S.; Schmidt, N.; Porter, D.; Jararweh, Y. A survey on blockchain for information systems management and security. *Inf. Process. Manag.* **2021**, *58*, 102397. [CrossRef]
20. Tanveer, M.; Bhutta, M.N.M.; Alzahrani, B.A.; Albeshri, A.; Alsubhi, K.; Chaudhry, S.A. CMAP-IoT: Chaotic Map-Based Authentication Protocol for Crowdsourcing Internet of Things. *Arab. J. Sci. Eng.* **2023**, 1–14. [CrossRef]
21. Tanveer, M.; Ahmad, M.; Nguyen, T.N.; Abd El-Latif, A.A. Resource-efficient authenticated data sharing mechanism for smart wearable systems. *IEEE Trans. Netw. Sci. Eng.* **2022**, *10*, 2525–2536. [CrossRef]
22. Petcu, A.; Pahontu, B.; Frunzete, M.; Stoichescu, D.A. A Secure and Decentralized Authentication Mechanism Based on Web 3.0 and Ethereum Blockchain Technology. *Appl. Sci.* **2023**, *13*, 2231. [CrossRef]
23. Ali, G.; Ahmad, N.; Cao, Y.; Khan, S.; Cruickshank, H.; Qazi, E.A.; Ali, A. xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things. *IEEE Access* **2020**, *8*, 58800–58816. [CrossRef]
24. Xiong, R.; Ren, W.; Hao, X.; He, J.; Choo, K.K.R. BDIM: A Blockchain-based Decentralized Identity Management Scheme for Large Scale Internet of Things. *IEEE Internet Things J.* **2023**. [CrossRef]
25. Ferlini, A.; Ma, D.; Harle, R.; Mascolo, C. EarGate: Gait-based user identification with in-ear microphones. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, New Orleans, LA, USA, 28 March–1 April 2022; pp. 337–349.
26. Marsico, M.D.; Mecca, A. A survey on gait recognition via wearable sensors. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–39. [CrossRef]
27. Wan, C.; Wang, L.; Phoha, V.V. A survey on gait recognition. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–35. [CrossRef]
28. Alessi, M.; Camillo, A.; Giangreco, E.; Matera, M.; Pino, S.; Storelli, D. Make users own their data: A decentralized personal data store prototype based on ethereum and ipfs. In Proceedings of the 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 26–29 June 2018; pp. 1–7.
29. Yuan, W.; Li, X.; Li, M.; Zheng, L. DCAGS-IoT: Dynamic Cross-Domain Authentication Scheme Using Group Signature in IoT. *Appl. Sci.* **2023**, *13*, 5847. [CrossRef]
30. Wu, L.; Yang, J.; Zhou, M.; Chen, Y.; Wang, Q. LVID: A multimodal biometrics authentication system on smartphones. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 1572–1585. [CrossRef]
31. Sengar, S.S.; Hariharan, U.; Rajkumar, K. Multimodal biometric authentication system using deep learning method. In Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 12–14 March 2020; pp. 309–312.
32. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772. [CrossRef]
33. Stylios, I.; Kokolakis, S.; Thanou, O.; Chatzis, S. Key factors driving the adoption of behavioral biometrics and continuous authentication technology: An empirical research. *Inf. Comput. Secur.* **2022**, *30*, 562–582. [CrossRef]
34. Labayen, M.; Vea, R.; Flórez, J.; Aginako, N.; Sierra, B. Online student authentication and proctoring system based on multimodal biometrics technology. *IEEE Access* **2021**, *9*, 72398–72411. [CrossRef]
35. Singla, S.K.; Singh, M.; Kanwal, N. Biometric system-Challenges and future trends. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 647–651.
36. Li, S.; Zhang, X.; Qian, Z.; Feng, G.; Ren, Y. Key based artificial fingerprint generation for privacy protection. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 828–840. [CrossRef]
37. Joseph, T.; Kalaiselvan, S.A.; Aswathy, S.U.; Radhakrishnan, R.; Shamna, A.R. RETRACTED ARTICLE: A multimodal biometric authentication scheme based on feature fusion for improving security in cloud environment. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 6141–6149. [CrossRef]
38. Cao, C.; Tang, Y.; Huang, D.; Gan, W.; Zhang, C. IIBE: An improved identity-based encryption algorithm for WSN security. *Secur. Commun. Netw.* **2021**, *2021*, 8527068. [CrossRef]
39. Rajasekar, V.; Premalatha, J.; Sathya, K. Enhanced biometric recognition for secure authentication using iris preprocessing and hyperelliptic curve cryptography. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8841021. [CrossRef]
40. Srinivas, J.; Das, A.K.; Wazid, M.; Kumar, N. Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 1133–1146. [CrossRef]
41. Khan, A.A.; Kumar, V.; Ahmad, M. An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 698–705. [CrossRef]
42. Lucia, O.; Isong, B.; Gasela, N.; Abu-Mahfouz, A.M. Device authentication schemes in IoT: A Review. In Proceedings of the IEEE 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 21–22 November 2019; pp. 1–6.
43. Melki, R.; Noura, H.N.; Chehab, A. Lightweight multi-factor mutual authentication protocol for IoT devices. *Int. J. Inf. Secur.* **2020**, *19*, 679–694. [CrossRef]
44. Lansky, J.; Sadrishojaei, M.; Rahmani, A.M.; Malik, M.H.; Kazemian, F.; Hosseinzadeh, M. Development of a lightweight centralized authentication mechanism for the Internet of Things driven by fog. *Mathematics* **2022**, *10*, 4166. [CrossRef]
45. Lee, J.H. BIDaaS: Blockchain based ID as a service. *IEEE Access* **2017**, *6*, 2274–2278. [CrossRef]

46. Zhang, L.; Li, H.; Sun, L.; Shi, Z.; He, Y. Poster: Towards fully distributed user authentication with blockchain. In Proceedings of the 2017 IEEE Symposium on Privacy-Aware Computing (PAC), Washington, DC, USA, 1–4 August 2017; pp. 202–203.
47. Al Omar, A.; Bhuiyan, M.Z.A.; Basu, A.; Kiyomoto, S.; Rahman, M.S. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. *Future Gener. Comput. Syst.* **2019**, *95*, 511–521. [CrossRef]
48. He, D.; Zeadally, S.; Xu, B.; Huang, X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2681–2691. [CrossRef]
49. Toutara, F.; Spathoulas, G. A distributed biometric authentication scheme based on blockchain. In Proceedings of the 2020 IEEE International Conference on Blockchain (Blockchain), Toronto, ON, Canada, 3–6 May 2020; pp. 470–475.
50. Das, A.K. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *Int. J. Commun. Syst.* **2017**, *30*, e2933. [CrossRef]
51. Dhar, S.; Khare, A. Blockchain-IoT based device authentication protocol for Healthcare Multimedia. In Proceedings of the IEEE 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 1–3 March 2023; pp. 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.